A Unified Span-Based Approach for Opinion Mining with Syntactic Constituents

Qingrong Xia[1], **Bo Zhang**[2], **Rui Wang**[3]*, **Zhenghua Li**[1]†, **Yue Zhang**[2], **Fei Huang**[2], **Luo Si**[2], **Min Zhang**[1]
[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, China
[2]Alibaba Group [3]Vipshop (China) Co., Ltd.
[1]kirosummer.nlp@gmail.com,{zhli13,minzhang}@suda.edu.cn
[2]{klayzhang.zb,shiyu.zy,f.huang,luo.si}@alibaba-inc.com
[3]mars198356@hotmail.com

Abstract

Fine-grained opinion mining (OM) has achieved increasing attraction in the natural language processing (NLP) community, which aims to find the opinion structures of "*Who expressed what opinions towards what*" in one sentence. In this work, motivated by its span-based representations of opinion expressions and roles, we propose a unified span-based approach for the end-to-end OM setting. Furthermore, inspired by the unified span-based formalism of OM and constituent parsing, we explore two different methods (multi-task learning and graph convolutional neural network) to integrate syntactic constituents into the proposed model to help OM. We conduct experiments on the commonly used MPQA 2.0 dataset. The experimental results show that our proposed unified span-based approach achieves significant improvements over previous works in the exact F1 score and reduces the number of wrongly-predicted opinion expressions and roles, showing the effectiveness of our method. In addition, incorporating the syntactic constituents achieves promising improvements over the strong baseline enhanced by contextualized word representations.

1 Introduction

Opinion mining (OM), which aims to find the opinion structures of "*Who* expressed *what opinions towards what*." in one sentence, has achieved much attention in recent years (Katiyar and Cardie, 2016; Marasović and Frank, 2018; Zhang et al., 2019b, 2020). The opinion analysis has many NLP applications, such as social media monitoring (Bollen et al., 2011) and e-commerce applications (Cui et al., 2017). The commonly used benchmark MPQA (Wiebe et al., 2005) uses span-based annotations to represent opinion expressions and roles. Figure 1 gives an example of its opinion structures with two opinion expressions and related roles.

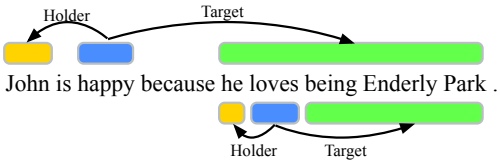


Figure 1: An example of OM, where the blue, yellow, and green blocks denote the opinion expressions, holders, and targets, respectively.

Previous OM works (Yang and Cardie, 2013; Katiyar and Cardie, 2016; Quan et al., 2019; Zhang et al., 2020) mainly treat it as a BMESO-style tagging problem, which converts opinion expressions and opinion roles (holder/target) into BMESO-based labels and uses a linking module to connect the predicted expressions and roles. The B, M, and E represent the beginning, middle, and ending word of a role, S denotes a single-word role, and O denotes other words. However, this kind of method is not perfect for the end-to-end OM setting, because one word can only belong to one opinion role (one word has only one label), while there exist overlapping opinion structures between different expressions in one sentence. Figure 1 gives an example, in which some overlapped opinion relations have been discarded by previous works (Katiyar and Cardie, 2016), such as [*happy*, *he loves being Enderly Park*, *Target*] and [*loves*, *he*, *Holder*]. There are also other works which focus only on predicting opinions roles based on the gold-standard expressions, which also follow the BMESO-based method (Marasović and Frank, 2018; Zhang et al., 2020). However, they also suffer from some weaknesses: 1) the expressions are usually fed into the model input as indicator embeddings (1 if the current word

* Rui Wang's contributions were carried out while at Alibaba Group.
† Corresponding author.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1795–1804
June 6–11, 2021.

belongs to an expression, 0 otherwise), thus one sample is expanded n times if one sentence has n expressions, which is inefficient (Marasović and Frank, 2018; Zhang et al., 2020). 2) The BMESO-based method is weak to capture long-range dependencies and prefers to predict shorter opinion role spans (Zhang et al., 2020).

Motivated by the span-based representations of opinion expressions and roles, we propose a unified span-based opinion mining model (SPANOM) that can solve or alleviate the aforementioned weaknesses. First, we treat the identification of opinion expressions and roles as two unified binary span classification problems, i.e., judging whether the word span is an expression (or role) or not. Then, we allocate the opinion relations on the predicted expression-role pairs. This strategy converts the overlapped opinion role identification of different expressions into classifying different expression-role pairs. For example, predicting [*happy*, *he loves being Enderly Park*, *Target*] and [*loves*, *he*, *Holder*] is infeasible in BMESO-based method, while it is feasible in our span-based method. Benefit from the model architecture, the proposed model only needs to train once for one sample in one epoch, which is very efficient for training. Besides, the unified model can be easily adapted to the given-expression setting by using gold-standard expressions. Furthermore, inspired by the same span-based formalism between the syntactic constituents and opinion roles, we explore two types of methods to encode the syntactic knowledge to improve the role spans recognition for two motivations, i.e., multi-task learning (MTL) for enhancing the model representative ability and graph convolutional networks (GCN) (Kipf and Welling, 2016; Guo et al., 2019) for encoding the constituent structures.

We conduct extensive experiments on the commonly used MPQA2.0 dataset and demonstrate that our proposed unified model achieves superior performance compared with previously proposed BMESO-based works. Our contributions are: (i) we propose a unified span-based model for opinion mining in the end-to-end fashion that also supports the given-expression setting, (ii) we successfully integrate syntactic constituents knowledge into our model with MTL and GCN, achieving promising improvements, (iii) detailed analyses demonstrate the effectiveness of our unified model and the usefulness of integrating constituent syntactic knowledge on the long-distance opinion roles.

2 Related Work

There are several task settings for opinion mining in the community: 1) Breck et al. (2007); Yang and Cardie (2014) focus on labeling the expressions. 2) Katiyar and Cardie (2016); Zhang et al. (2019b); Quan et al. (2019) discover the opinion structures in the end-to-end setting, i.e, based on the systematic expressions. 3) Marasović and Frank (2018); Zhang et al. (2019a, 2020) identify the opinion roles based on the given expressions. Our work follows the end-to-end setting and also supports the given-expression setting.

Most of the previous opinion mining works treat it as a BMESO-tagging problem, which can be handled by the typical sequence labeling model, such as bi-directional long-short term memory network conditional random field (BiLSTM-CRF). Yang and Cardie (2013) propose to use traditional feature-based CRF model to predict the BMESO-based opinion role labels. Katiyar and Cardie (2016) propose a BiLSTM-CRF model to first predict the word-wise opinion role label and then determine the relationship with the expression by the role label and distance to the expressions. Zhang et al. (2019b) propose a transition-based model for opinion mining, which identifies opinion expressions and roles by the human-designed transition actions. Quan et al. (2019) integrate BERT representations into a BiLSTM-CRF model, but they do not distinguish different expressions in one sentence. As aforementioned, it is trivial for the sequence labeling style models to handle the overlapped opinion roles belonging to different expressions in one sentence.

Due to the issue of data scarcity, several kinds of external knowledge have been investigated to improve OM performance. Marasović and Frank (2018) propose several MTL frameworks with semantic role labeling (SRL) to utilize semantic knowledge. Zhang et al. (2019a) extract the semantic representations from a pre-trained SRL model and feed them into the opinion mining model, achieving substantial improvements. Zhang et al. (2020) incorporate the powerful contextual representations of bi-directional encoder representations from Transformers (BERT) (Devlin et al., 2019) and external dependency syntactic knowledge.

To solve or alleviate the weaknesses of the previously proposed BMESO-based models, we propose a new method to unifiedly model the opinion expressions and roles, which treats the expres-

sion identification, role identification, and opinion relation classification as an MTL problem. Besides, to boost the opinion mining performance and motivated by the span-based task formalism, we explore to incorporate syntactic constituents into our model. Utilizing span-based representations have been investigated for many other NLP tasks, such as named entity recognition (NER) (Tan et al., 2020), constituency parsing (Kitaev and Klein, 2018), and semantic role labeling (SRL) (He et al., 2018). Generally, NER is a single span classification problem, constituency parsing is a span-based structure prediction problem, and SRL is a word-span classification problem. Different from them, in our methodology, OM is a span-span classification problem.

3 The SpanOM Model

3.1 Task Definition.

Given an input sentence $s = w_1, w_2, ..., w_n$, our model aims to predict the gold-standard opinion structures $\mathcal{Y} \subseteq \mathcal{E} \times \mathcal{O} \times \mathcal{R}$, where $\mathcal{E} = \{w_i, ..., w_j | 1 \leq i \leq j \leq n\}$ is the set of *expressions*, $\mathcal{O} = \{w_i, ..., w_j | 1 \leq i \leq j \leq n\}$ is the set of opinion *roles* , and $\mathcal{R}$ is the set of opinion relations (*holder* and *target*) with a dummy relation ψ that represents no relation.

Accordingly, we treat the opinion expression and role recognition as the unified span classification problem and determine the opinion relation based on the predicted expressions and roles. We jointly model the three sub-tasks in an MTL fashion to enhance the modules' interplay. The left part of Figure 2 shows the model architecture of our model and we will detailedly describe the components in the following sections.

3.2 Input Layer.

For each word w_i in sentence s, we employ word embedding, char representation, and contextual word representation to compose the model input, denoted as:

$$\mathbf{x}_i = \mathbf{emb}^{\texttt{word}}_{w_i} \oplus \mathbf{rep}^{\texttt{char}}_{w_i} \oplus \mathbf{rep}^{\texttt{context}}_{w_i|s}, \quad (1)$$

where $\oplus$ means the concatenate operation. We use the convolutional neural networks (CNN) (Kalchbrenner et al., 2014) to generate the character representations over the characters of words.

3.3 Encoder Layer.

Over the input layer, we employ BiLSTM to encode the model input. We treat the concatenation of the outputs of the left-to-right LSTM and right-to-left LSTM as the output:

$$\begin{aligned} \overrightarrow{\mathbf{h}}_i &= \overrightarrow{LSTM}(\mathbf{x}_i, \mathbf{h}_{i-1}), \\ \overleftarrow{\mathbf{h}}_i &= \overleftarrow{LSTM}(\mathbf{x}_i, \mathbf{h}_{i+1}), \\ \mathbf{h}_i &= \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i. \end{aligned} \quad (2)$$

3.4 Span Representation and Identification Layer.

To better distinguish opinion expression and role representations, we first employ two multi-layer perceptions (MLP) to re-encode the output of BiLSTM encoder, denoted as:

$$\mathbf{h}^{\texttt{exp}}_i = MLP^{exp}(\mathbf{h}_i), \mathbf{h}^{\texttt{rol}}_i = MLP^{rol}(\mathbf{h}_i). \quad (3)$$

For a word span that begins at b-th word and ends at e-th word, we define it as $\texttt{span}_{b,e}$. So the representations of expression and role are defined as:

$$\begin{aligned} \mathbf{span}^{\texttt{exp}}_{b,e} &= (\mathbf{h}^{\texttt{exp}}_b + \mathbf{h}^{\texttt{exp}}_e) \oplus (\mathbf{h}^{\texttt{exp}}_b - \mathbf{h}^{\texttt{exp}}_e), \\ \mathbf{span}^{\texttt{rol}}_{b,e} &= (\mathbf{h}^{\texttt{rol}}_b + \mathbf{h}^{\texttt{rol}}_e) \oplus (\mathbf{h}^{\texttt{rol}}_b - \mathbf{h}^{\texttt{rol}}_e). \end{aligned} \quad (4)$$

Given the representations of expressions and roles, we employ another two MLPs to classify whether the span is the gold expression/role or not. Furthermore, we also incorporate the span boundary information to help the determination of spans. Specifically, we employ another four MLPs on the span boundary positions to determine whether the word is a boundary position or not[1]. Thus, the score formulation of the span is as:

$$\begin{aligned} \mathbf{s}^{\texttt{exp}} &= MLP^{\texttt{exp}}(\mathbf{span}^{\texttt{exp}}_{b,e}) \\ &\quad + MLP^{\texttt{exp}}_b(\mathbf{h}_b) + MLP^{\texttt{exp}}_e(\mathbf{h}_e), \\ \mathbf{s}^{\texttt{rol}} &= MLP^{\texttt{rol}}(\mathbf{span}^{\texttt{rol}}_{b,e}) \\ &\quad + MLP^{\texttt{rol}}_b(\mathbf{h}_b) + MLP^{\texttt{rol}}_e(\mathbf{h}_e). \end{aligned} \quad (5)$$

We can observe that for a sentence with n words, the numbers of candidate spans for expressions and roles are both $\frac{n*(n+1)}{2}$, while the number of gold expressions and roles are much fewer. To alleviate the unbalanced number of gold samples

[1] We omit the process of span boundary module in Figure 2 for clarity.

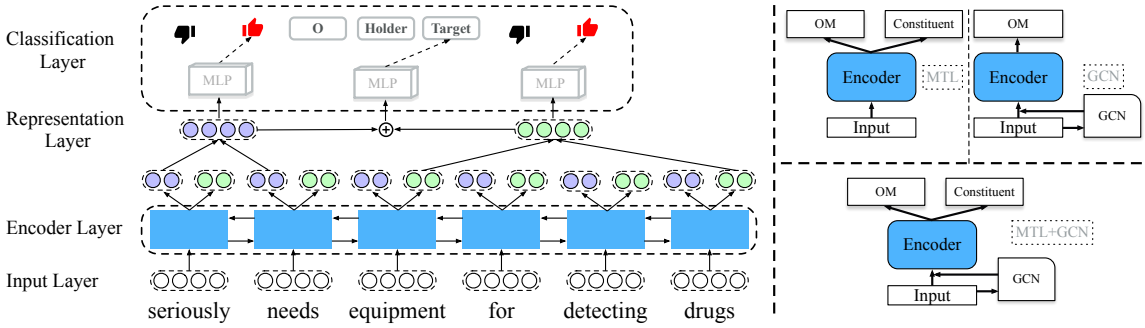


Figure 2: The model architecture of our unified span-based opinion mining model (left) and syntactic constituent integration methods (right).

and negative samples, we adapt the focal loss that is widely used in computer vision (Lin et al., 2017) into our model. Formally, for every span i in a sentence, the sentence focal loss is defined as:

$$Loss = -\sum_{i}\sum_{c}(1-p_{i,c})^{\gamma}y_{i,c}log(p_{i,c}), \quad (6)$$

where $p_{i,c}$ is the softmax value of the $\mathbf{s}_c^{\texttt{exp}}$ (or $\mathbf{s}_c^{\texttt{opi}}$) for class c of span i, γ is a pre-defined hyper-parameter and $y_{i,c}$ is an indicator value that equals to 1 if c is the ground-truth class 0 otherwise. Compared with the typical cross-entropy loss, the difference appears in the first item, which can intuitively make the model focus more on the hard-to-classify samples. We denote the loss of the opinion expressions and roles as $L^{\texttt{exp}}$ and $L^{\texttt{rol}}$, respectively.

3.5 Relation Classification Layer.

Given the predicted opinion expressions and roles, the next step is to determine the opinion relation (holder, target, or no relation) for each expression-role pair. We employ another MLP classifier to compute the score for each relation of the focused expression $\mathbf{span}^{\texttt{exp}}$ and role $\mathbf{span}^{\texttt{rol}}$:

$$\mathbf{s}^{\texttt{rel}} = MLP(\mathbf{span}^{\texttt{exp}} \oplus \mathbf{span}^{\texttt{rol}}). \quad (7)$$

Focal loss is also employed to estimate this module, which is denoted as $L^{\texttt{rel}}$.

3.6 Training and Inference.

We sum the three losses from the three modules as the final model loss:

$$L^{\texttt{OM}} = L^{\texttt{exp}} + L^{\texttt{rol}} + L^{\texttt{rel}}. \quad (8)$$

For the end-to-end OM setting, the model predicts the relation of the predicted expressions and roles. As for the given-expression mode, we directly feed the gold expressions into the model, with other parts the same as the end-to-end mode. During the inference process, we employ dynamic programming to predict opinion expressions and roles.

4 Syntactic Constituents

Since the data scale is relatively small, previous works usually try to integrate external knowledge to enhance the basic OM model and improve its performance (Marasović and Frank, 2018; Zhang et al., 2019a). Previous sequence tagging models usually incorporate word-wise external information, such as dependency parsing (Zhang et al., 2020). We try to investigate the integration of constituent knowledge, which is motivated by their unified span-based formalism. Two different methods are explored in our work, i.e., MTL and GCN.

4.1 The MTL Method.

MTL is an effective method to utilize external knowledge, which is usually by sharing the model parameters of the main task and auxiliary task (Ruder, 2017). Considering the efficiency of full constituent parsing, we use partial constituent parsing in our model, i.e., training partial constituent trees (constituent spans), not the entire constituent tree. In detail, we first extract all the constituent spans[2] from the OntoNotes corpus. See 5.1 for the detailed settings. Then, we add a span classification module over the BiLSTM encoder, which is similar to the unified opinion classifier, to predict the span belonging to which kind of constituent labels. Third, with the addition of the constituent span classification module, we can easily allocate automatic constituent labels to enhance the predicted opinion expressions and roles. Thus, we create randomly initialized constituent label embeddings for representing the syntactic labels, which are then

[2]We remove constituent spans with label "Top" and "S".

concatenated with the expression and role representations:

$$\begin{aligned}\mathbf{span}_{b,e}^{\texttt{exp}'} &= \mathbf{span}_{b,e}^{\texttt{exp}} \oplus \mathbf{emb}_{\texttt{exp}}^{label}, \\ \mathbf{span}_{b,e}^{\texttt{rol}'} &= \mathbf{span}_{b,e}^{\texttt{exp}} \oplus \mathbf{emb}_{\texttt{rol}}^{label}.\end{aligned} \tag{9}$$

The syntax-enhanced span representations are then passed to participate in the later computation process. Finally, the focal loss is used to estimate the partial constituent tree prediction module and the partial constituent loss ($L^{\texttt{cons}}$) is used to update the shared input layer, encoder layer, and the partial constituent parsing classification layer. So the loss of our constituents-enhanced OM model becomes:

$$L = L^{\texttt{OM}} + \alpha L^{\texttt{cons}}. \tag{10}$$

It is worth noting that the data size of OM and constituent trees is different, so we employ a corpus-weighting parameter α to balance it. In general, the MTL method brings two benefits: 1) enhancing the model encoder and 2) adding constituency label information to expressions and roles.

4.2 The GCN Method.

The MTL method enhances our OM model from the aspect of model representative ability by jointly modeling opinion mining and partial constituency parsing. We argue that modeling the syntactic constituent structure is also beneficial for OM because it provides valuable syntactic information for a sentence. Therefore, we try to employ the recently popular GCN (Kipf and Welling, 2016) to encode the constituent structure. However, the conventional GCN is not suitable for constituency trees, because it usually works on the dependency trees (Zhang et al., 2018, 2020) where the nodes are the surface words in a sentence. While, in constituent trees, there exists a certain number of non-terminal nodes[3], such as "NP", "VP", "SBAR" and so on. So it is hard to directly apply conventional GCN on the constituent trees. In the following, we first introduce the definition and workflow of typical GCN and then describe our modification.

Formally, we denote an undirected graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the set of nodes and edges, respectively. The GCN computation flow of node $v \in \mathcal{V}$ at l-th layer is defined as:

$$\mathbf{h}_v^l = \rho\left(\sum_{u \in \mathcal{N}(v)} \mathbf{W}^l \mathbf{h}_u^{l-1} + \mathbf{b}^l\right), \tag{11}$$

where $\mathbf{W}^l \in \mathrm{R}^{m \times m}$ is the weight matrix, $\mathbf{b}^l \in \mathrm{R}^m$ is the bias term, $\mathcal{N}(v)$ is the set of all one-hop neighbour nodes of v, and ρ is an activation function (relu activation function in our work). Especially, $\mathbf{h}_u^0 \in \mathrm{R}^m$ is the initial input representation, and m is the representation dimension.

Since there are some non-terminal nodes in the constituent tree, the GCN input can not directly get from the surface words. We create a randomly initialized non-terminal embedding matrix $\mathbf{E}^{N \times D}$ and a dynamic mask for composing the GCN input and extracting the GCN output, where N is the number of non-terminal nodes and D is the dimension of the terminal node inputs. There are two main ways to add the GCN modules in the neural network models, i.e., concatenating with the input layer and stacking over the encoder layer. According to our preliminary experiments, we choose the former method. In detail, we treat the composition of non-terminal node representations and terminal node representations as the GCN input, and then concatenate the terminal node GCN outputs $\mathbf{x}_i^{\text{GCN}}$ with the basic model input as the final model input. The top right part of Figure 2 shows the overall workflow.

The final constituent-enhanced unified span-based opinion mining model combines the two methods, which we denoted as "MTL+GCN" in the later sections. The workflow is shown by the right bottom part of Figure 2.

5 Experiments

5.1 Settings.

We conduct experiments on the commonly used English MPQA2.0 dataset (Wiebe et al., 2005). Following the data split of previous works (Zhang et al., 2019a, 2020), the development data contains 132 documents and the test data contains 350 documents, using five-fold cross-validation to evaluate the test data. For constituent data, we use the OntoNotes 5.0 dataset (Pradhan et al., 2013) in our MTL method. We use the constituent parser of Kitaev and Klein (2018)[4] to obtain the automatic constituent trees. BERT (Devlin et al., 2019) is employed as the external contextual representations. We implement our model with Pytorch[5] and the basic model has 20.46M parameters[6].

[3] Terminal nodes are the surface words in the sentence.

[4] The parser achieves 93.55 F1 score on the PTB development data.

[5] https://pytorch.org/

[6] We release the code, configurations, and models at https://github.com/KiroSummer/opinion_mining_with_syn_cons.

Models	Exact F1			Binary F1			Proportional F1		
	Holder	Target	Overall	Holder	Target	Overall	Holder	Target	Overall
Katiyar and Cardie (2016)	-	-	-	58.22	54.98	-	-	-	-
Zhang et al. (2019b)	47.02	31.45	-	60.93	56.44	-	-	-	-
Quan et al. (2019)+BERT	-	-	-	55.52	50.39	-	46.62	34.29	-
SPANOM	52.90	32.42	43.12	56.47	45.09	51.04	55.62	41.65	48.90
SPANOM+BERT	58.24	41.10	**49.89**	62.04	53.27	57.76	61.20	49.88	55.68

Table 1: Experimental results of our span-based opinion mining model and comparison with previous works on the MPQA2.0 dataset in the end-to-end setting. "-" means results are not reported in their paper.

Models	Exact		
	P	R	F1
Zhang et al. (2019b)	60.21	48.52	53.04
SPANOM	64.85	52.60	58.06
SPANOM+BERT	67.15	60.63	**63.71**

Table 2: Results and comparison of the expression prediction on the exact metric in the end-to-end setting.

5.2 Hyper-parameters.

We employ the 300-dimension GloVe vector (Pennington et al., 2014) as our pre-trained word embeddings. The character embeddings are randomly initialized and a CNN with kernel sizes of 3, 4, 5 is used to capture the character representations. For the contextual representations, we extract the representations from the base BERT by making a weighted summation over the last four layer outputs. The hidden size of the BiLSTM layer is set to 300 and we employ 2-layer BiLSTMs to encode the input representations. The dimension of opinion expression and role representations is 300 and the hidden size of expression, role, and relation classifiers is 150. We use 3-layer GCNs with hidden size 300. The dropout rate of the input layer, encoder layer, and other components are 0.5, 0.4, and 0.3, respectively. The hyper-parameter γ is 3.0.

5.3 Training Criterion.

We employ Adam optimizer with an L2 weight decay of 1e-6 to optimize our model. The batch size is 32. The initial learning rate is set to 0.001 and decays 0.99 for every 50 steps. Our model trains for at most 320k steps and early stops if no performance gains happen in 100 epochs on the development data. We pick the model that performs best on the development data for evaluation. It costs about 4 minutes to run one epoch training and 1 minute for evaluation.

5.4 Evaluation Metrics.

Following previous works (Marasović and Frank, 2018; Zhang et al., 2020), we use the Precision, Recall, and F1 score to measure the experimental results regarding to *Exact* match setting, and two other auxiliary metrics of *Binary* and *Proportional* match. The average value of the five-fold cross-validation results is reported in our work. The binary and proportional metrics are also called *overlap* metric, which includes the opinion roles that exactly match the gold opinions and inexactly match but overlap with gold roles. In detail, the binary match means an opinion overlaps with a gold-standard opinion and the proportional match computes the maximum ratio value of an role with the overlapped gold role.

5.5 Results of SPANOM.

Results in the end-to-end setting. Table 1 lists the results of previous works and our model (SPANOM) in the end-to-end setting. First, our model achieves superior performance than previous works in terms of exact F1 score, reaching better results of 52.90 and 32.42 exact F1 scores on the *holder* and *target* roles. The overall exact F1 score of the two roles is 43.12. Second, integrating BERT representations into the model input can bring substantial improvements, achieving 49.89 exact F1 score. We can see that in the auxiliary metrics of binary and proportional, previous works perform better than ours, which we think because our model more focuses on the entire word spans and we will detailedly discuss it in the analysis section. Finally, the results of expression prediction are shown in Table 2. We can see that our model outperforms Zhang et al. (2019b) by +5.02 exact F1 score.

Results in the given-expression setting. Table 3 shows the experimental results and comparison with previous works in the given-expression set-

Models	Exact F1			Binary F1			Proportional F1		
	Holder	Target	Overall	Holder	Target	Overall	Holder	Target	Overall
Zhang et al. (2019a)	73.07	42.70	58.30	81.57	68.34	75.15	79.35	61.22	70.55
Zhang et al. (2020)	73.05	44.21	58.79	81.21	69.50	75.43	79.33	62.53	71.03
Zhang et al. (2020)+BERT	76.74	52.61	64.73	85.45	75.74	80.62	83.58	69.31	76.48
SPANOM	72.40	45.83	59.62	78.10	64.51	71.56	76.74	58.74	68.08
SPANOM+BERT	76.47	54.95	**65.95**	82.69	72.93	77.93	81.53	67.42	74.64

Table 3: Experimental results of our span-based opinion mining model and comparison with previous works on the MPQA2.0 dataset in the given-expression setting.

Models	Exact F1		
	Holder	Target	Overall
end-to-end setting			
SPANOM+BERT	58.24	41.10	49.89
SPANOM+BERT+SYNCONS	58.46	41.82	**50.46**
given-expression setting			
Marasović and Frank (2018)+SRL	75.58	46.40	61.51
Zhang et al. (2019a)+SRL	76.95	50.50	63.74
Zhang et al. (2020)+BERT	76.74	52.61	64.73
Zhang et al. (2020)+BERT+SYNDEP	79.51	56.61	**68.08**
SPANOM+BERT	76.47	54.95	65.95
SPANOM+BERT+SYNCONS	78.34	56.96	68.02

Table 4: Experimental results of our model with external syntactic knowledge and comparison with previous works. "SYNCONS" means "MTL+GCN".

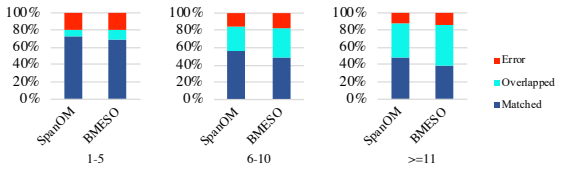


Figure 3: Percentage comparison of the "matched", "overlapped", and "error" predicted opinion roles of the outputs from the SPANOM model and BMESO-based model on the entire test data.

ting. First, we can see that our proposed span-based model outperforms previously proposed BMESO-based models in the exact F1 score metric, achieving 59.62 exact F1 score. Second, when using contextual word representations of BERT, our model consistently outperforms the previous best result, resulting in a new state-of-the-art result of 65.95 exact F1 score, showing superior performance compared with the BMESO-based methods.

5.6 Results of Integrating Syntactic Constituents.

Table 4 shows the results of our model integrating syntactic constituents and compare with previous works with SRL or dependency syntax knowledge. In the end-to-end setting, incorporating constituent knowledge brings an improvement of +0.57 exact F1 score. In the given-expression setting, we can see that integrating constituent syntactic knowledge into our model brings a +2.07 exact F1 score improvement, achieving comparable results with previous best results of Zhang et al. (2020). Even though our basic OM model outperforms Zhang et al. (2020), the improvements from syntactic constituents lag behind the dependency syntax. We think this is partly because of the relatively low performance of constituent parsing (93.55 F1 score) compared with dependency parsing (95.7 F1 score). Apart from syntactic knowledge, Marasović and Frank (2018); Zhang et al. (2019a) both try to encode semantic knowledge, but their models don't use BERT representations.

6 Analysis

In this section, we conduct detailed analyses to gain more insights into our unified OM model and the effectiveness of integrating syntactic constituents.

6.1 Span-based Model vs. BMESO-based Model

As the experimental results shown, our span-based model performs better in the exact matching metric than the BMESO-based models, while the BMESO-based models have better results in the auxiliary overlap metric. To understand the performance difference, we list the detailed percentage of opinion statistics of the system outputs of our span-based model and the BMESO-based model of Zhang et al. (2019a) in Figure 3, both using the BERT representations. The "Matched", "Overlapped" and "Error" mean the predicted opinion role matches the gold role, *not matches but overlaps part of the gold role* and totally mismatches the gold role, respectively. We can see that: 1) our model achieves better per-

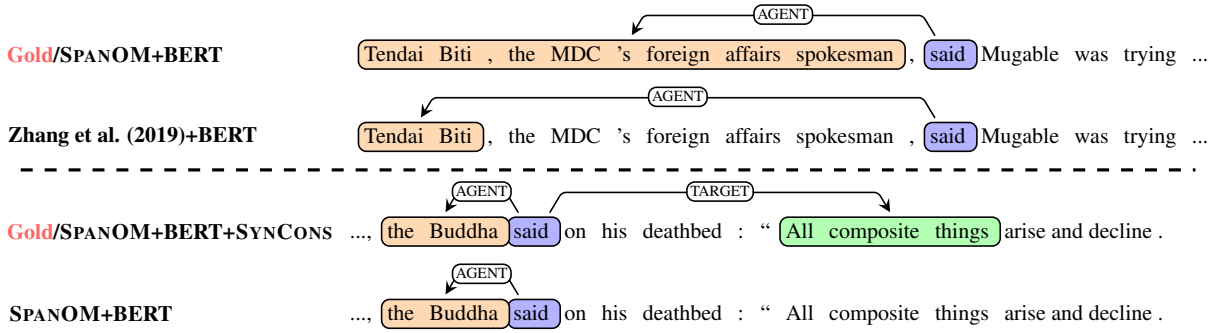


Figure 4: Examples of case study, where the upper example shows the comparison of our span-based model with the previous BMESO-based model of Zhang et al. (2019a) and the bottom example shows the comparison of our BERT-based model with/without syntactic constituents.

formance on the exact match setting through all the span length scenarios, especially on the spans that contain more than 10 words, 2) the BMESO-based model outputs more overlapped opinion roles than our span-based model, thus the BMESO models have better results in the auxiliary metric of binary and proportional settings. This demonstrates that our SpanOM more focuses on the full opinion role spans while the BMESO-based method may weak to give high exact predictions.

Case study. The upper part of Figure 4 shows an example of the output of our span-based model and previous BMESO-based model of Zhang et al. (2019a). We can find that the span-based model successfully predicts the full agent while the BMESO-based model only predicts part of the agent span. This confirms the intuition that our span-based model is more good at predicting the long-range arguments, while the BMESO-based model is weak at long-range spans, which is consistent with the findings of Zhang et al. (2020).

6.2 Effect of Syntactic Constituents

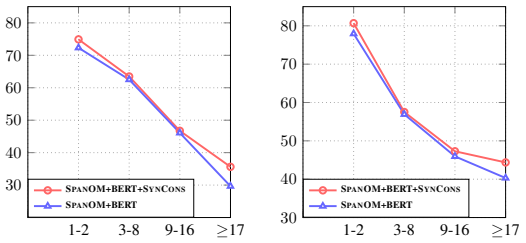
Which source of constituent knowledge is better? There are two main constituent syntax corpus in the community, i.e., Penn Treebank (PTB) (Marcus et al., 1993) and OntoNotes5.0 (Weischedel et al., 2013). The PTB corpus contains about 39k training data and mainly focuses on news data, while the OntoNotes5.0 corpus contains about 75k training data and focuses on multi-domain data (news, web, telephone conversation, and etc.).

It is a worthy question to explore which is better for our span-based OM model, or what kind of combination is better. We compare them with various combinations on the BERT-based model, whose results are shown in Table 5. First, the second major row shows the results of our model with the MTL method, where MTL with PTB achieves the best exact F1 score of 68.02. Second, the results of our model with the GCN method are listed in the third major row, where "OntoNotes" and "PTB" means the automatic constituent trees are generated by parser trained on OntoNotes[7] and PTB, respectively. We can see that using the automatic constituent trees from $\texttt{Parser}^{\texttt{PTB}}$ achieves the best exact F1 score of 67.66. Finally, we try to combine the two kinds of methods and the results are shown in the last major row. It is clear that combining the MTL method with OntoNotes and the GCN method with $\texttt{Parser}^{\texttt{PTB}}$ achieves better results than the reversed one. Therefore, our constituent-enhanced opinion mining model follows this combination. Besides, we can also see the relative lower results of "OntoNotes+PTB" in "+MTL" and "+GCN" settings, which is strange

Models		Dev (F1)		
		Exact	Binary	Prop
SpanOM+BERT		66.64	77.27	74.41
+MTL	OntoNotes	67.72	78.30	75.74
	PTB	68.02	77.68	75.61
	OntoNotes+PTB	67.24	77.70	75.61
+GCN	OntoNotes	66.77	76.73	74.30
	PTB	67.66	77.48	75.24
	OntoNotes+PTB	67.65	78.08	75.76
+MTL&GCN	PTB&OntoNotes	67.21	77.96	75.23
	OntoNotes&PTB	**68.55**	77.61	75.62

Table 5: The performance of different kinds of constituents knowledge on the first folder data of MPQA2.0 in the given-expression setting. "Prop" means proportional. "A+B" means combining the two corpus and "A&B" means using corpus A for the MTL method and automatic trees from $\texttt{Parser}^{\texttt{B}}$ for the GCN method.

[7] We use the code of Kitaev and Klein (2018) to train the OntoNotes constituent parser, which achieves 92.20 F1 score on the development data.



(a) Opinion Length (b) Distance to Expressions

Figure 5: Exact F1 score of our model with/without syntactic knowledge regarding to different role length and the distance from expression to roles.

that combining more information leads to lower performance. We think this is mainly caused by the different domains of the data in OntoNotes. As is well known, learning uniform knowledge from different domains data is a challenging problem. So, in the MTL method, adding OntoNotes into PTB can enhance such domain problems, and vice versa. In the GCN method, the two GCN outputs are concatenated, so the potential conflicts of different arcs are alleviated. Thus, the performance didn't drop too much.

We also try to utilize dependency syntax. However, it brings less improvement compared with constituent syntax, which is understandable that word-based information is not very appropriate for the span-based model. It is also consistent with our intuition that span-based syntactic constituents are more suitable for the span-based model.

Why and where do syntactic constituents help? OM aims to discover the structure of "Who expressed what" in a sentence and constituent syntax provides valid information like the "NP" and "VP" phrases in a sentence. Intuitively, the "agent/target and expression" may be covered by "NP and VP" phrases. We make statistics on the overlapping of constituent spans and opinions. We find that about 88% opinion roles can be covered by the predicted constituent spans from the MTL module, where the most four are "NP", "VP", "SBAR" and "PP". Since the constituent knowledge can intuitively help the determination of roles, we list the result of the different span lengths in Figure 5a. We can find that *constituent knowledge helps most on those opinion roles with longer length*. We also report the results regarding the distance between the expressions and roles in Figure 5b, which shows a similar conclusion.

Case study. The bottom part of Figure 4 gives a case study that shows the difference between syntax-enhanced and syntax-agnostic models. We can see that the target argument "*All composite things*" is hard to be identified by our baseline model. When integrating constituent knowledge, the model correctly discovers this opinion role and give the "target" relation. We think it is because the constituent tree gives a "NP" label to the word span, which helps our model to identify it. We also observe that there are some peculiarities of the MPQAs annotation scheme. For example, in the sentence "*The criteria set by Rice are the following: the three countries in question are repressive ...*", "*set by*" is the expression, "*Rice*" is the holder, and "*the three countries in question*" is the target. However, "*set by*" is not a constituent phrase at all. In fact, "*by*" and "*Rice*" compose a prepositional phrase in the constituent tree. So, it is hard for our model to recognize "*set by*" as an opinion expression. Besides, "*the three countries in question*" is also not a dependent of the opinion expression "*set by*", in which the constituent tree can not provide valuable structural information for the two phrases. Such phenomena is hard to handle by our model and raise challenges to the future work.

7 Conclusion

In this paper, we propose a unified span-based opinion mining model that can handle the overlapped opinion roles, providing a new methodology. Our proposed model outperforms previously proposed BMESO-based models in terms of exact match metric on both the end-to-end and given-expression settings. Furthermore, integrating syntactic constituents knowledge with MTL and GCN brings substantial improvements over our BERT-enhanced baseline model. Detailed analyses show the difference between the span-based model and the BMESO-based model and the effectiveness of incorporating syntactic constituents on the determination of opinion role spans.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China (Grant No. 62036004, 61876116), a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and was partially supported by Alibaba Group through Alibaba Research Intern Program.

References

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *IJCAI*, volume 7, pages 2683–2688.

Lei Cui, Shaohan Huang, Furu Wei, Chuanqi Tan, Chaoqun Duan, and Ming Zhou. 2017. Superagent: A customer service chatbot for e-commerce websites. In *Proceedings of ACL 2017, System Demonstrations*, pages 97–102.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association for Computational Linguistics*, 7:297–312.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proceedings of ACL*, pages 919–929.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of ACL*, pages 2676–2686.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ana Marasović and Anette Frank. 2018. SRL4ORL: Improving opinion role labeling using multi-task learning with semantic role labeling. In *Proceedings of NAACL-HLT*, pages 583–594.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of CoNLL*, pages 143–152.

Wei Quan, Jinli Zhang, and Xiaohua Tony Hu. 2019. End-to-end joint opinion role labeling with bert. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2438–2446. IEEE.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. 2020. Boundary enhanced neural span classification for nested named entity recognition. In *AAAI*, pages 9016–9023.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of ACL*, pages 1640–1649.

Bishan Yang and Claire Cardie. 2014. Joint modeling of opinion expression extraction and attribute classification. *Transactions of the Association for Computational Linguistics*, 2:505–516.

Bo Zhang, Yue Zhang, Rui Wang, Zhenghua Li, and Min Zhang. 2020. Syntax-aware opinion role labeling with dependency graph convolutional networks. In *Proceedings of ACL*, pages 3249–3258.

Meishan Zhang, Peili Liang, and Guohong Fu. 2019a. Enhancing opinion role labeling with semantic-aware word representations from semantic role labeling. In *Proceedings of NAACL-HLT*, pages 641–646.

Meishan Zhang, Qiansheng Wang, and Guohong Fu. 2019b. End-to-end neural opinion extraction with a transition-based model. *Information Systems*, 80:56–63.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*, pages 2205–2215.